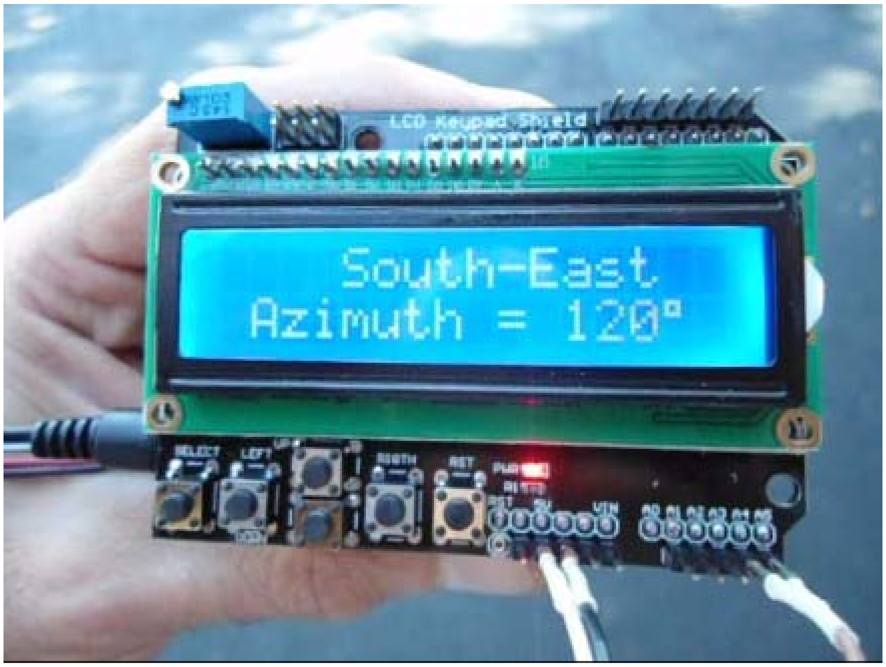
ARDUINO LCD COMPASS

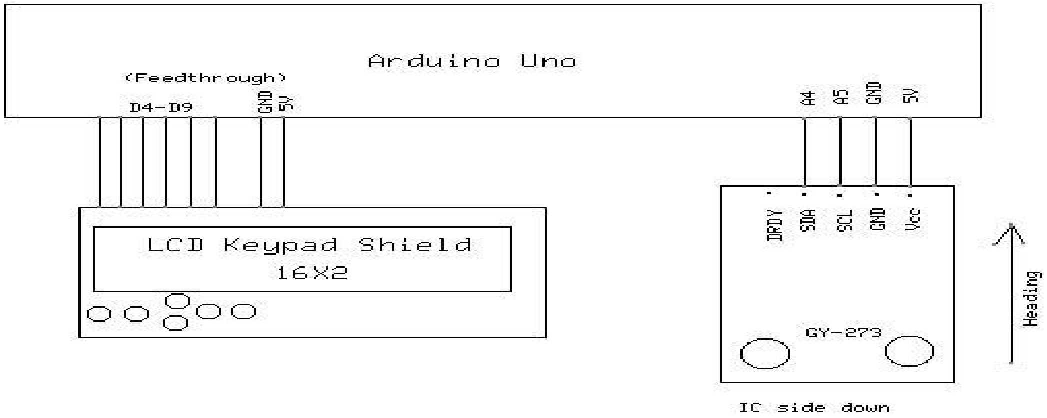
Nick Cinquino 10/4/2015

A well functioning and easy-to-build LCD display compass can be made with a minimum of parts, in a short period of time.
The circuit displays azimuth or heading in degrees, along with 8-point compass directions. The parts required are an Arduino Uno microcontroller, an LCD Keypad Shield, a GY-273 magnetic field sensor breakout board, and 4 cables with female Dupont connectors.

Assembly is simple: solder pin headers to the Keypad Shield, to at least analog pins 4 and 5, and to +5V and ground on the Keypad Shield. Plug the Arduino into the Keypad Shield. Add cables to A4, A5, +5V and GND, and connect to the GY-273 breakout as shown in the schematic. The cables should be at least 6" long to keep the sensor as far away from the Arduino and Keypad Shield as possible. Copy and paste the following sketch into your Arduino program window. Verify, save, and download to the Arduino. No special library is required.



Display on the LCD Keypad shield, with Azimuth degrees and 8-point direction.



How it works: The sensor (HMC5883) outputs 3 axes of orthogonal magnetic field data. We collect 2 of them (X and Y) and calculate azimuth from the Arc Tangent of X,Y, times (180/pi)+180 to take it from radians to degrees. The azimuth in degrees is sent to the LCD, along with the determined 8-point compass direction.

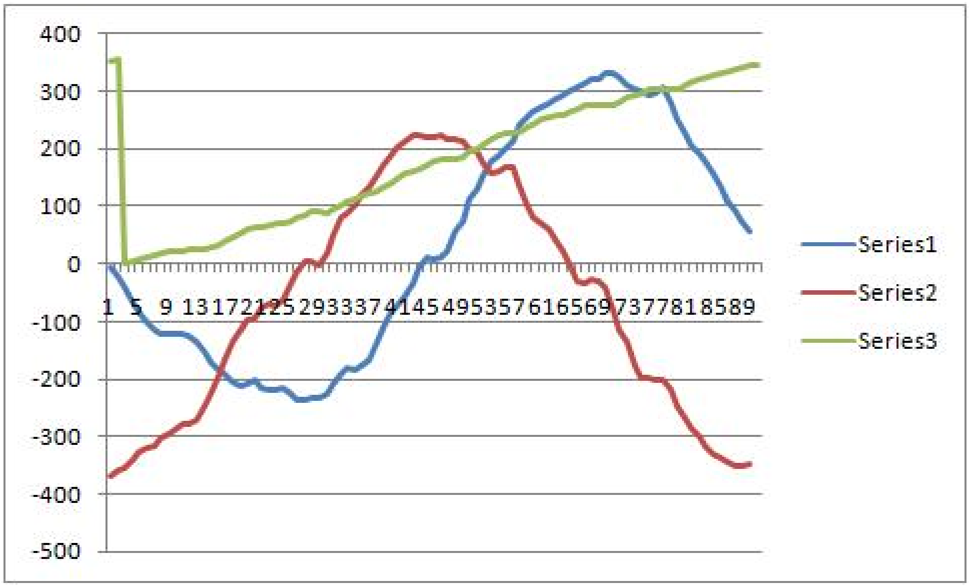
Parts sources: The GY-273 is available from Hobbycomponents.com, as well as Amazon, and Aliexpress. The LCD Keypad shield and Arduino are from many sources including Arduino.cc,
Amazon, etc. The cable was homemade but the 20cm female to female Dupont jumpers from Aliexpress or DX.com would be fine.

The sketch was modified from:
http://forum.hobbycomponents.com/viewtopic.php?f=73&t=1485

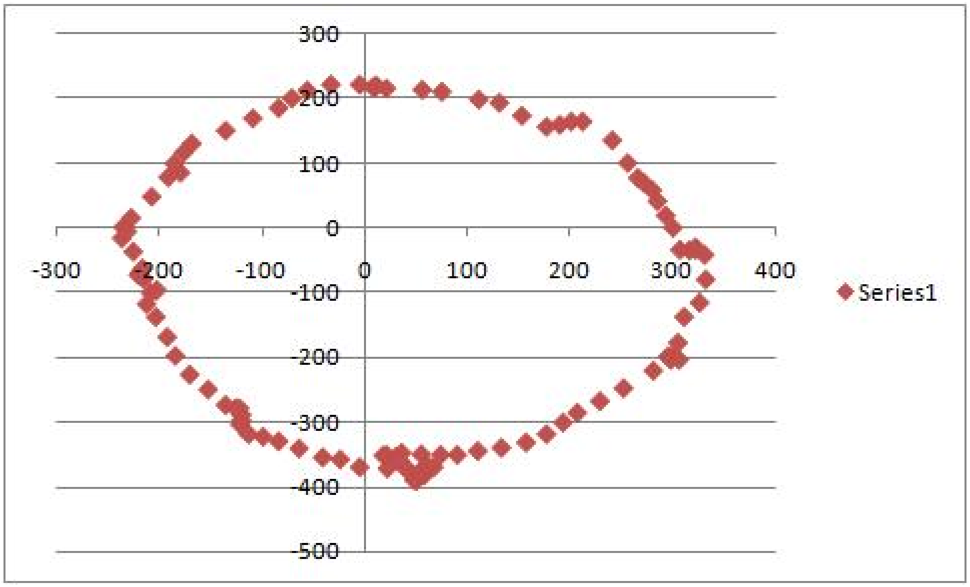
Complete Arduino Sketch follows:

```
/*
CONNECTIONS:
MODULE    ARDUINO
VCC       3.3V
GND       GND
SCL       A5
SDA       A4
DRDY      No connection
 */


#include <Wire.h>  /* Include the standard Wire library */
#include <LiquidCrystal.h>  //LCD library
LiquidCrystal lcd(8,9,4,5,6,7);  //LCD pinout
#define HMC5803L_Address 0x1E  /* The I2C address of the module */
#define X 3   /* Register address for the X Y and Z data */
#define Y 7
#define Z 5
double angle;


  void setup()
{
  lcd.begin(16, 2);
  Serial.begin(9600);
  Wire.begin();  /* Initialise the Wire library */
  Init_HMC5803L();   /* Initialise the module */
}


void loop()
{
  angle= (atan2((double)HMC5803L_Read(X),(double)HMC5803L_Read(Y)) * (180 / 3.14159265)) + 180; // angle in degrees

  Serial.print(HMC5803L_Read(X));   /* Read each sensor axis data and output to the serial port */
  Serial.print(", ");
  Serial.print(HMC5803L_Read(Y));
  Serial.print(", ");
  Serial.print(HMC5803L_Read(Z));
  Serial.print(", ");
  Serial.print(angle,0);
  Serial.print(", ");

  lcd.clear();
  lcd.setCursor(4,0);

  if((angle <= 22.1) || (angle >= 337.1 )){
        Serial.print("North");
        lcd.print("North");}
    if((angle >= 22) && (angle <=67.1 )){
        Serial.print("North-East");
        lcd.print("North-East");}
    if((angle >= 67) && (angle <= 112.1 )){
        Serial.print("East");
        lcd.print("East");}
    if((angle >= 112) && (angle <= 157.1 )){
        Serial.print("South-East");
        lcd.print("South-East");}
    if((angle >= 157) && (angle <= 202.1 )){
        Serial.print("South");
        lcd.print("South");}
    if((angle >= 202) && (angle <= 247.1 )){
        Serial.print("South-West");
        lcd.print("South-West");}
    if((angle >= 247) && (angle <= 292.1 )){
        Serial.print("West");
        lcd.print("West");}
    if((angle >= 292) && (angle <= 337 )){
        Serial.print("North-West");
        lcd.print("North-West");}

  Serial.println(" ");

  lcd.setCursor(1,1);
  lcd.print("Azimuth = ");
  lcd.print(angle,0);
  lcd.print((char)223);


  delay(250);   /* Wait a little before reading again */
}


/* This function will initialise the module and only needs to be run once
   after the module is first powered up or reset */
void Init_HMC5803L(void)
{
  /* Set the module to 8x averaging and 15Hz measurement rate */
  Wire.beginTransmission(HMC5803L_Address);
  Wire.write(0x00);
  Wire.write(0x70);

  /* Set a gain of 5 */
  Wire.write(0x01);
  Wire.write(0xA0);
  Wire.endTransmission();
}
```

```
/* This function will read once from one of the 3 axis data registers
and return the 16 bit signed result. */
int HMC5803L_Read(byte Axis)
{
  int Result;

  /* Initiate a single measurement */
  Wire.beginTransmission(HMC5803L_Address);
  Wire.write(0x02);
  Wire.write(0x01);
  Wire.endTransmission();
  delay(6);

  /* Move modules the resiger pointer to one of the axis data registers */
  Wire.beginTransmission(HMC5803L_Address);
  Wire.write(Axis);
  Wire.endTransmission();

  /* Read the data from registers (there are two 8 bit registers for each axis) */
  Wire.requestFrom(HMC5803L_Address, 2);
  Result = Wire.read() << 8;
  Result |= Wire.read();

  return Result;
}
```



Plotting the X and Y separately in Excel, while turning full circle. The green line is degrees. Ideally, peak amplitudes would be equal.



XY scatter plot of the same data in Excel. Ideally it should be a near-perfect circle.

References: There are 2 documents from Honeywell, a manufacturer of magnetoresistive sensors including the one in use, that were very helpful in putting the circuit and programming together into a working system. They are:

https://aerospace.honeywell.com/~/media/Images/Plymouth%20Website%20PDFs/Magnetic%20Sensors/Technical%20Articles/Applications_of_Magnetic_Sensors_for_Low_Cost_Compass_Systems.ashx

https://aerospace.honeywell.com/~/media/Images/Plymouth%20Website%20PDFs/Magnetic%20Sensors/Technical%20Articles/Applications_of_Magnetoresistive_Sensors_in_Navigation_Systems.ashx